|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | |
| **Department of Computer Science and Engineering (AIML)** | | | | | |
| Subject name: Machine Learning | | | | | | Subject Code: CSL604 | |
| Class | | TE CSE | | | Semester –VI (CBCGS) | Academic year: 2024-25 | |
| Name of Student | |  | | | | **QUIZ Score :6** | |
| Roll No | |  | | Experiment No. | | 09 | |
| Title: **To perform Principal Component Analysis (PCA).** | | | | | | | |
|  | | | | | | | |
| **1. Lab objectives applicable:**  **LOB1:**To introduce platforms such as Anaconda, COLAB suitable to Machine Learning.  **LOB3:** To develop Neural Network based learning models | | | | | | | |
| **2. Lab outcomes applicable:**  **LO4: To** apply Dimensionality Reduction techniques. | | | | | | | |
| **3. Learning Objectives:**   1. To perform and learn the principal component analysis. | | | | | | | |
| **4. Practical applications of the assignment/experiment:**  To perform dimensionality reduction. | | | | | | | |
| **5. Prerequisites**:   1. Python language | | | | | | | |
| **6. Minimum Hardware Requirements**:-  I series processor, RAM 4GB,  **7. Software Requirements:-**  Colab or Visual Studio or Jupyter notebook (Anaconda) | | | | | | | |
| **8. Quiz Questions :** [**https://docs.google.com/forms/d/e/1FAIpQLSdVetsa67X9Cnbp1N64KnljBFnvDsoVSCZWtzLfnKTxgGme6g/viewform?usp=dialog**](https://docs.google.com/forms/d/e/1FAIpQLSdVetsa67X9Cnbp1N64KnljBFnvDsoVSCZWtzLfnKTxgGme6g/viewform?usp=dialog) | | | | | | | |
| **9. Experiment Evaluation:** | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | **Marks obtained** | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | 6 | 6 |
| **2** | Lab Performance | | | | |  | 2 |
| **3** | Punctuality | | | | |  | 2 |
| **Date of performance (DOP)** | | |  | | **Total marks obtained** |  | **10** |

**Signature of Faculty**

**10. Theory:**

Input Array=

Step 1:Mean Centering

Mean(X1) =

Mean(X2) =

Mean Centering:

Step 2: Compute CoVariance Matrix

Step 3: Compute Eigen Value and Eigen Vectors

Solving these equations we get;

Solving these equations we get;

Step 4: Sort Eigenvalues and Select Principal Component

Step 5:Obtain

**11. Installation Steps / Performance Steps and Results –**

**Source Code:**

import numpy as np

import matplotlib.pyplot as plt

# Sample dataset (2D)

X = np.array([[2.5, 2.4],

              [0.5, 0.7],

              [2.2, 2.9],

              [1.9, 2.2],

              [3.1, 3.0],

              [2.3, 2.7],

              [2.0, 1.6],

              [1.0, 1.1],

              [1.5, 1.6],

              [1.1, 0.9]])

# Step 1: Mean Centering

mean\_X = np.mean(X, axis=0)

X\_centered = X - mean\_X

# Step 2: Compute Covariance Matrix

cov\_matrix = np.cov(X\_centered.T)

# Step 3: Compute Eigenvalues and Eigenvectors

eigenvalues, eigenvectors = np.linalg.eig(cov\_matrix)

# Step 4: Sort Eigenvalues and Select Principal Component

sorted\_indices = np.argsort(eigenvalues)[::-1]  # Sort in descending order

principal\_component = eigenvectors[:, sorted\_indices[0]]  # First eigenvector

# Step 5: Project Data onto Principal Component

X\_pca = X\_centered @ principal\_component  # 1D Projection

# Step 6: Reconstruct 2D points from 1D projection for visualization

X\_reconstructed = np.outer(X\_pca, principal\_component) + mean\_X

# ---- Plot the Original Data, Principal Component, and Projected Data ----

plt.figure(figsize=(8, 6))

# Scatter original data points

plt.scatter(X[:, 0], X[:, 1], color='blue', label='Original Data')

# Draw principal component line

pc\_line = np.array([mean\_X - 3 \* principal\_component, mean\_X + 3 \* principal\_component])

plt.plot(pc\_line[:, 0], pc\_line[:, 1], 'k--', label='Principal Component')

# Scatter projected points (onto principal component)

plt.scatter(X\_reconstructed[:, 0], X\_reconstructed[:, 1], color='red', label='Projected Data')

# Connect original points to their projections

for i in range(len(X)):

    plt.plot([X[i, 0], X\_reconstructed[i, 0]], [X[i, 1], X\_reconstructed[i, 1]], 'gray', linestyle='dotted')

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

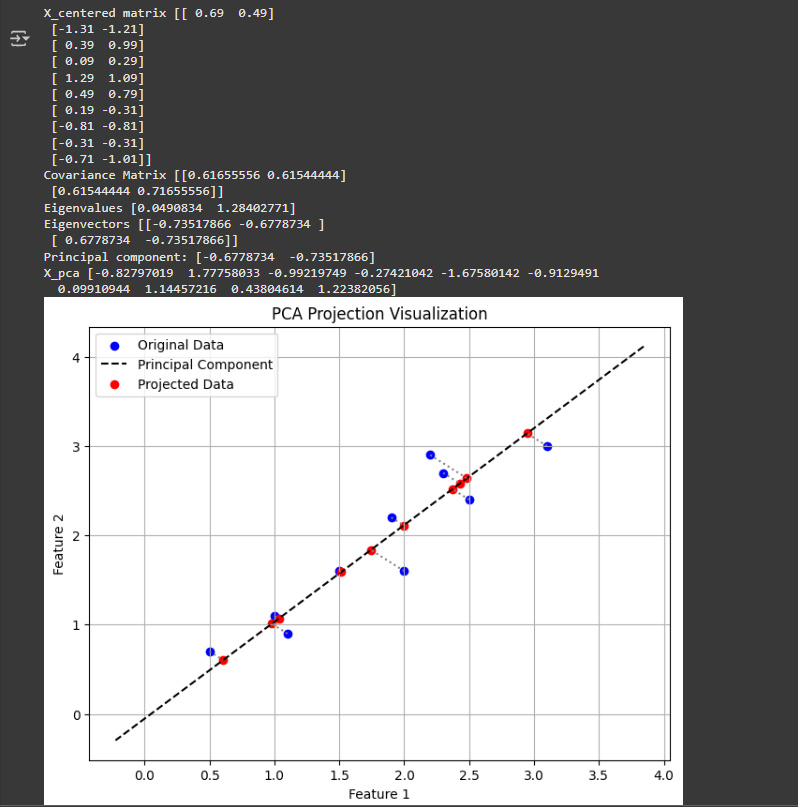
plt.title("PCA Projection Visualization")

plt.legend()

plt.grid()

plt.show()

**Output:**



**12. Learning Outcomes Achieved**

1. Students are able to perform dimensionality reduction by using PCA.

**13. Conclusion:**

**1. Applications of the Studied Technique in Industry**

PCA is widely used in image compression, gene expression analysis, and financial risk modeling to reduce data complexity while preserving critical information. Industries leverage it for face recognition, anomaly detection, and recommendation systems to enhance efficiency and performance.

**2. Engineering Relevance**

PCA is crucial in data preprocessing, noise reduction, and feature selection, making it essential for engineers working in machine learning, signal processing, and big data analytics. It helps optimize computational resources while improving model accuracy.

**3. Skills Developed**

This experiment enhances mathematical reasoning through eigen decomposition and variance analysis. It strengthens coding skills in implementing dimensionality reduction techniques and builds analytical thinking for handling high-dimensional datasets efficiently.

**14. References**:

1. Nathalie Japkowicz & Mohak Shah, ―Evaluating Learning Algorithms: A Classification Perspective‖, Cambridge.
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖
3. Samir Roy and Chakraborty, ―Introduction to soft computing‖, Pearson Edition.
4. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher

Education

1. Peter Flach, ―Machine Learning‖, Cambridge University Press
2. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
3. Kevin P. Murphy, ―Machine Learning ― A Probabilistic Perspective‖, MIT Press